

AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method for managing a plurality of processors as a virtual devices, said method comprising:

receiving a device request at a task queue manager running on a first processor in a computer system, wherein the computer system includes a plurality of heterogeneous processors that share a common memory and wherein the device request corresponds to a virtual device;

storing data corresponding to the request in the common memory;

identifying a second processor from the plurality of processors to handle the request, wherein the first processor and the second processor are dislike processors, wherein the first processor executes a first instruction set and wherein the second processor executes a second instruction set;

creating, by the first processor, a task block in the common memory, the task block including a software code identifier and an input buffer address;

signaling, from the first processor, the identified second processor, wherein the signaling includes writing the address of the task block to a mailbox corresponding to the second processor;

~~receiving the data corresponding to the request at the second processor; and~~
receiving, at the second processor, the address of the task block from the second processor's mailbox;

retrieving, at the second processor, the software code identifier from the task block;

reading data from an input buffer located in the common memory at a location corresponding to the input buffer address into the second processor's local memory, wherein the reading is performed using a direct memory access (DMA) operation;

determining whether the software code corresponding to the software code identifier is loaded in the second processor's local memory; and
in response to determining that the software code corresponding to the software code identifier is not loaded in the second processor's local memory:

reading the software code from the common memory into the second processor's local memory, wherein the reading is performed using a DMA operation; and

processing the data by the second processor using the software code stored in the second processor's local memory.

2. (canceled)
3. (canceled)
4. (canceled)
5. (canceled)
6. (canceled)
7. (canceled)
8. (canceled)
9. (canceled)
10. (canceled)
11. (canceled)
12. (canceled)
13. (canceled)
14. (canceled)
15. (canceled)
16. (canceled)
17. (canceled)
18. (canceled)
19. (canceled)
20. (canceled)

21. (canceled)
22. (canceled)
23. (canceled)
24. (canceled)
25. (canceled)
26. (canceled)
27. (canceled)
28. (canceled)
29. (canceled)
30. (canceled)
31. (canceled)
32. (canceled)
33. (canceled)
34. (canceled)
35. (canceled)
36. (canceled)
37. (canceled)
38. (canceled)
39. (canceled)
40. (canceled)
41. (canceled)
42. (canceled)
43. (canceled)
44. (canceled)
45. (canceled)
46. (canceled)
47. (canceled)
48. (canceled)
49. (canceled)

- 50. (canceled)
- 51. (canceled)
- 52. (canceled)
- 53. (canceled)
- 54. (canceled)
- 55. (canceled)
- 56. (canceled)
- 57. (canceled)
- 58. (canceled)
- 59. (canceled)
- 60. (canceled)